Robust Multi-Level Video Representation Using Mean Shift Analysis

Hai Gao*, Xiao-Dong Yu*, Lei Wang*, Ping Xue*, and Qi Tian†

*School of EEE, Nanyang Technological University, Singapore 639798.
†Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613.
*{ehgao, exdyu, elwang, epxue} @ntu.edu.sg, †tian@i2r.a-star.edu.sg

Abstract

A robust method for multi-level video representation based on the mean shift analysis (MSA) of low-level visual features is proposed in this paper. By tuning the bandwidth of MSA, video representation from the coarse level to the fine level can be achieved. This representation form provides a flexible scheme for content-based video analysis such as summarization, classification, and retrieval. Compared with the conventional k-means or fuzzy c-means algorithms, our method can adjust the resolution of representation in a more straightforward way, and is more robust since it does not need to initialize the cluster centers.

1. Introduction

In video analysis, a video sequence is usually modeled as a set of points in the feature space spanned by low-level visual features such as color, objects, motion, etc [1-3]. Video representation is then defined as the description of these points.

Due to the large amount of video data, a compact, informative video representation is ideal for fast and convenient browsing, indexing and retrieval. Nevertheless, compact and informative are two conflicting demands and it is a tough problem to find a trade-off between them: too compact representation faces the risk of losing some important information while too informative representation will result in redundancy. Thus, multi-level video representation becomes a practical choice.

In literature, the algorithms for multi-level video representation can be classified into two categories: hierarchical clustering [4] [2] and non-hierarchical clustering [5] [6]. For hierarchical clustering, Uchihashi et al. [4] reported a bottom-up iterative merging approach to build a hierarchical representation for video summary; Kobla et al. [2] applied the curve simplification method to extract key-frames at different level of details. For non-hierarchical clustering, Hanjalic and Zhang [5] proposed a video abstraction scheme based on k-means clustering and cluster-validity analysis, the video abstraction can be illustrated in a multi-level structure when there are multiple clustering options; Ferman and Tekalp [6] employed the fuzzy-c means clustering algorithm and incorporated the user’s preferences to produce multi-level video summary.

Because a matrix of distances (similarities) does not have to be determined, and basic data do not have to be stored during the computer run, the non-hierarchical methods are more suitable for large video data sets than the hierarchical techniques [7]. However, for non-hierarchical methods, both k-means clustering [5] and fuzzy c-means clustering [6] are sensitive to the initialization of the cluster centers. In addition, it is difficult to adjust the resolutions in these algorithms.

Mean shift analysis (MSA) [8] is an efficient approach for feature space analysis and it has been widely used in recent years. Comaniciu et al. used MSA for image segmentation [8] and object tracking [9]. Georgescu et al. used MSA for texture classification [10]. DeMenthon et al. used MSA for video object segmentation [11].

In this paper, we proposed to explore MSA to achieve multi-level video representation by tuning the bandwidth of MSA. For a given bandwidth, our method classifies all the frames into a number of clusters, and the cluster centers (which correspond to the key-frames) are produced automatically, no other extra efforts are needed. Compared with the methods proposed in [5] and [6], our method is more robust since it does not need to initialize the cluster centers and it is more straightforward to adjust the resolutions.

The remainder of this paper is organized as follows. Section 2 describes the principles of the mean shift analysis. Section 3 proposes a scheme for multi-level video representation (MLVR) and key-frame extraction based on MSA. Experimental results are shown in Section 4. Some conclusions are given in the last section.

2. Principles of Mean Shift Analysis
2.1. Non-parametric Density Estimation

Let \( x_1, x_2, \ldots, x_n \) be a set of \( n \) points in the \( d \)-dimensional space. The density at a point \( x \) (\( x \) is a \( d \)-dimensional vector) is often estimated locally by a number of neighboring samples in a window around \( x \) as follows [8]:

\[
\hat{p}_x(x) = \frac{1}{n \cdot h^d} \sum_{i=1}^{n} k \left( \frac{x - x_i}{h} \right)
\]

where \( k(u) \) is a kernel function [8], and \( h \) is called the window width, or the bandwidth.

Figure 1 illustrates the influence of the bandwidth on the estimated density function in the one-dimensional space. With different bandwidths, we will get estimated density functions with different resolutions.

![Figure 1. Estimated density functions with different bandwidths: the bandwidth used in (b) is larger than the bandwidth used in (a).](image)

2.2. Sample Mean Shift

Denote \( g(x) = -K'(x) \) and define a kernel \( G \) as \( G(x) = cg \left( \|x\|_2^2 \right) \) where \( c \) is a normalization constant which makes \( G(x) \) satisfy \( \int G(x) dx = 1 \). It can be proven [8] that

\[
\frac{\nabla \hat{p}_x(x)}{\hat{p}_x(x)} = \frac{2}{c} \sum_{i=1}^{n} g \left( \frac{x - x_i}{h} \right) \left( \frac{x_i}{h} \right)
\]

Denote the term in the bracket as \( M_{h,G}(x) \), it is called the sample mean shift vector at point \( x \).

2.3. Mean Shift Algorithm

Expression (2) shows that the sample mean shift vector obtained with Kernel \( G \) is an estimate of the normalized density gradient obtained with Kernel \( k \). Note that \( \hat{p}_x(x) > 0 \), so the gradient vector \( \nabla \hat{p}_x(x) \) is in the same direction as the sample mean shift vector \( M_{h,G}(x) \).

**Mean Shift Algorithm:** letting

\[
x_{i+1} = x_i + M_{h,G}(x_i)
\]

transform each observation recursively as follows:

\[
x_{i+1} = x_i + \left( \sum_{k=1}^{n} \frac{k}{h} \right) \left( \frac{x_i - x_k}{h} \right)
\]

The mean shift algorithm shifts each observation by the mean shift vector which is in the same direction as the gradient vector at the observation point. Thus after each iteration, each observation will have moved closer to its cluster center.

To simply illustrate this kind of moving, suppose the points are one-dimensional. In this case, the density gradient \( \nabla \hat{p}_x(x) \) at point \( x \) is also one-dimensional. Figure 2 shows this kind of moving. For more details, we consider a point \( x_1 \) in Figure 2: let \( x_1^0 = x_1 \), since \( \nabla \hat{p}_x(x) \mid_{x=x_1^0} > 0 \), \( x_1^1 \) will move to the right of \( x_1^0 \); for the second iteration, if \( \nabla \hat{p}_x(x) \mid_{x=x_1^1} > 0 \), \( x_1^2 \) will move to the right of \( x_1^1 \), otherwise it will move to the left of \( x_1^1 \); and so on. This is the analog of the linear iteration technique for stepping into the roots of the equation

\[
\nabla \hat{p}_x(x) = 0
\]

and equivalently the cluster centers. Although local minima are also roots of (5), it is easy to see that the algorithm (4) will move observations away from these points since the gradients at these points point away from them.

![Figure 2. Moving closer to the cluster centers after each iteration.](image)

Comaniciu [8] proved that sequences (4) are convergent, and they converge to their cluster centers.

3. Multi-Level Video Representation and Key-frame Extraction with MSA

Considering the capability of MSA aforementioned, we apply MSA to the video domain to achieve the proposed MLVR scheme.

Firstly, we extract the color histograms from the MPEG video as the feature. The color histograms in our algorithm are constructed from the DC-image, which is recovered from the MPEG stream by Yeo’s method [12]. It is composed of 24 bins, in which 16 for \( Y \) component and 4 for \( C_b \) and \( C_r \) components respectively. For the \( i \)-th frame, the color histogram \( H_i \) is a 24-dimensional vector as follows:

\[
H_i = [h_i(0), h_i(1), \ldots, h_i(15), h_i(0), h_i(0), h_i(0), h_i(2)],
\]

\[
h_i(3), h_i(5), h_i(0), h_i(2), h_i(3)]
\]

Suppose there are \( n \) frames in a video, the histograms will form a data set of 24 columns and \( n \) rows. We first do a preprocessing on this data set. We
calculate the standard variance $\sigma^2$ for each column $j$, and then delete those columns with small $\sigma^2$.

After this preprocessing, we form an $n \times d$ data set $h_{i,j}$ ($i = 1, 2, \cdots, n; j = 1, 2, \cdots, d$) where $d \leq 24$. This $n \times d$ data set is further normalized to 0 and 65535 as follows:

$$x_{i,j} = \frac{h_{i,j} - \text{minVal}}{\text{maxVal} - \text{minVal}} \times 65535,$$

where minVal and maxVal are the minimum and maximum value of the $d \times n$ data set respectively.

Thus we form $n$ points $x_1, x_2, \cdots, x_n$. Each point represents a frame and is a $d$-dimensional vector. MSA is then performed in this $d$-dimensional space.

In the $j$-th iteration of a point $x_j$ (see Expression (6)), we need to find all the points which are in the window around the point $x_j$. In fact, for mean shift analysis, the most expensive operation is finding the closest neighbors of a point in the space. A simple and efficient algorithm for nearest neighbor search in high dimensions [13] is employed in our implementation.

To define the distance between two points, the $L_1$ norm is used in this paper, i.e., the length of a vector $(a_1, a_2, \ldots, a_d)$ is defined as $(|a_1| + |a_2| + \cdots + |a_d|)$.

The mean shift iteration moves points towards cluster centers. When the bandwidth $h$ is very small, for example, every point does not have neighbors within distance $(h \times d$ for the $L_1$ norm), in this case, each point will be moved to itself, so there are $n$ cluster centers and each cluster contains only one point. As $h$ increases, those clusters that are close to each other will be merged and the number of the clusters will be reduced. The larger the bandwidth $h$, the less number of clusters the points will be classified into. In this strategy, we can provide a multi-resolution representation of the points, i.e., we can provide a multi-level representation of videos.

Besides that, another magical advantage of our method is that no other extra efforts are needed for extraction of key-frames which are of great importance for video content analysis such as summarization, classification, and retrieval. Note that the MSA moves points towards cluster centers and classifies all the points into a set of clusters. For each cluster, the original point (frame) which is closest to the cluster center is selected as the key-frame.

4. Experimental Results

4.1. Mean Shift Analysis on Synthetic Data

Normally, the video feature space is high dimensional, and it is not easy to illustrate the MSA with figures. To better illustrate the MSA, a synthetic data set containing 3500 points in a 2-dimensional space was generated as follows: first we generate 7 points

$$(x_i, y_i)(0 \leq x_i \leq 1; 0 \leq y_i \leq 1; i = 1, 2, \cdots, 7)$$

randomly as cluster centers. For each cluster, 500 points are generated, and the distances between the points and the cluster center are normally distributed with $\sigma^2=0.06$. The original data set is shown in Figure 3(a). The data set is first normalized to 0 and 65535 and then MSA is performed thereafter.

![Figure 3. Illustration of the MSA with different bandwidth $h$. (The small square boxes represent the cluster centers.)](image)

Figure 3 illustrates the MSA on the synthetic data set with different bandwidth $h$. When $h$ is 2000, the data set is classified into 7 clusters (each cluster is represented by a color). As $h$ increases to 5000, 7000, and 9000, the data set is classified into 5, 4, and 3 clusters, respectively. Thus we get a multi-level representation of the original data set.

4.2. Mean Shift Analysis on Video Data

We use the video clip “Apollo” to demonstrate our MLVR method. The clip begins with the astronauts getting off the bus and approaching the launch tower (shot 1: 1-616), and then the astronauts take the lift in the tower (shot 2: 617-1319) and walk into the spacecraft (shot 3: 1320-1534) followed by a shot about the audience (shot 4: 1535-1662). And then a shot about the commander (shot 5: 1663-1832), a wide-angle shot of the command hall (shot 6: 1833-1960).

Using the color histograms as described in Section 3 for mean shift analysis, when $h=2000$, the frames are classified into 6 clusters, the cluster centers (k-frames) are 282, 906, 1380, 1606,1727, and 1848. As $h$ increases, those clusters close to each other are merged
as illustrated in Figure 4, thus producing a multi-level representation of the “Apollo” video clip.

![Images](image1.png)

Figure 4. MSA on the “Apollo” video clip.

Table I shows the experimental results with other video clips.

<table>
<thead>
<tr>
<th>Clip Name</th>
<th>Shot Number</th>
<th>h=2000</th>
<th>h=3000</th>
<th>h=5000</th>
<th>h=8000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siege</td>
<td>19</td>
<td>21</td>
<td>18</td>
<td>16</td>
<td>13</td>
</tr>
<tr>
<td>Homeroad</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>News</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>

Figure 5 illustrates the key-frames extracted with the k-means clustering algorithm proposed by Hanjalic and Zhang [5]. It shows that the results in different runs are inconsistent and sensitive to the initializations.

![Images](image2.png)

Figure 5. Key-frames extracted with k-means clustering (k=5). Each row corresponds to results from a computer run.

5. Conclusions

We have proposed a novel method for multi-level video representation based on the mean shift analysis of low-level visual features. The experimental results showed the effectiveness of our method. It not only characterizes the video content from the coarse level to the fine level, but also generates the key-frames automatically. Compared with the conventional k-means or fuzzy c-means algorithms, our method is more robust and it can adjust the resolution of representation in a more straightforward way.

6. Acknowledgment

This work is supported by A*STAR, Singapore under Grant LIT 2002-4.

7. References